

AMENDMENTSIn the Claims

1. (Currently Amended) ~~In a system management controller included in a computer system, a~~ A method of accessing event data describing a failure, ~~the method~~ comprising:
providing a computer system with a system management controller coupled between a processor bus and a local bus;
configuring the system management controller to monitor a task of writing data to an event log, the task being executed by a Basic Input Output System (BIOS) program in response to the failure;
monitoring the task for completion;
accessing the event data if the task fails to complete; and,
writing the event log via the system management controller in response to accessing the event data.
2. (Original) The method of claim 1, wherein the failure generates a system management interrupt and the BIOS program is triggered in response to the system management interrupt.
3. (Original) The method of claim 1, wherein monitoring the task comprises:
setting a configurable time of a watchdog timer, the task being configured to access the event data, and write the data to the event log in response to the event data, the task being completed within the configurable time set in the watchdog timer;
receiving an indication from the BIOS program on completion of the task.
4. (Original) The method of claim 3, wherein the task fails to complete when the task fails to receive the indication from the BIOS program.
5. (Original) The method of claim 3, wherein receiving the indication from

the BIOS program comprises resetting the configurable time in the watchdog timer.

6. (Original) The method of claim 1, wherein the event data is stored in a memory of the computer system by a controller device included in the computer system.

7. (Original) The method of claim 6, wherein the controller device is a memory controller.

8. (Original) The method of claim 6, wherein the controller device is an I/O controller.

9. (Original) The method of claim 1, wherein the system management controller accesses the event data over a system bus of the computer system.

10. (Original) The method of claim 9, wherein the system bus is a SMBus.

11. (Original) The method of claim 1 further comprising, the system management controller writing the event log in response to accessing the event data.

12. (Previously Presented) The method of claim 1, wherein writing the event log occurs over a system bus of the computer system.

13. (Original) The method of claim 12, wherein the system bus is a SMBus.

14. (Currently Amended) A method of accessing event data on a failure of a computer system, the method comprising:

executing a BIOS program to access the event data in response to a first failure of the computer system;

triggering a watchdog timer in a system management controller of the computer system, the watchdog timer being triggered substantially concurrent to the first failure, the system management controller being coupled between a processor bus and a local bus;

configuring the watchdog timer to allow the BIOS program to complete in absence of a second failure;
determining whether the execution of the BIOS program caused the second failure, the second failure forcing the watchdog timer to expire; and
the system management controller accessing the event data when the watchdog timer expires; and,
writing the event log via the system management controller in response to accessing the event data.

15. (Original) The method of claim 14, wherein the second failure is substantially similar to the first failure.

16. (Original) The method of claim 14, wherein the second failure occurs while a processor included in the computer system operates in a SMM mode.

17. (Currently Amended) A computer system comprising:
a processor;
a memory coupled to the processor;
a BIOS program stored in the memory, the BIOS program being operable to write data to an event log in response to a critical event;
a system controller coupled to the memory and the processor, the system controller being coupled between a processor bus and a local bus, the system controller operable to:
receive an indication of the critical event;
upon receipt of the indication, initiate operation of a timer;
determine whether the BIOS program has written the data to the event log within a configurable period of time defined by the timer; and,
write the data to the event log if the BIOS program has not written the data to the event log.

18. (Currently Amended) In a computer system having a processor and a system controller, the system controller coupled between a processor bus and a local bus,

a method of responding to an event, the method comprising:

issuing an interrupt to the processor in response to the event;
detecting the interrupt at the system controller coupled to the processor;
initiating a timer in the system controller upon detection of the interrupt;
attempting to write data to an event log by executing a BIOS program;
the system controller determining whether the execution of the BIOS program
resulted in writing data to the event log; and,
writing data to the event log via the system management controller if the
execution of the BIOS program did not result in writing data to the event
log.

19. (Original) The method of claim 18 further comprising:

if execution of the BIOS does not result in the writing of the data to the event log
before expiration of a time period established by the timer, causing the
system controller to respond to the event.

20. (Currently Amended) A method for accessing and writing event data to a
log on failure of a computer system including a system management controller, the
system management controller coupled between a processor bus and a local bus, the
method comprising:

monitor a task of writing data to an event log via the system management
controller, the task being executed by a Basic Input Output System (BIOS)
program in response to the failure;
monitoring the task for completion to determine whether the BIOS program was
able to complete writing the data to the event log;
accessing the event data if the task failed to complete; and,
writing the event log via the system management controller if the task failed to
complete.

21. (Previously Presented) The method of claim 20, wherein if the task failed
to complete, the system management controller generates a system management interrupt
and the BIOS program is triggered in response to the system management interrupt.

22. (Previously Presented) The method of claim 20, wherein monitoring the task comprises:

setting a configurable time of a watchdog timer, the task being configured to access the event data, and write the data to the event log in response to the event data, the task being completed within the configurable time set in the watchdog timer;

receiving an indication from the BIOS program on completion of the task.

23. (Previously Presented) The method of claim 22, wherein the task fails to complete when the task fails to receive the indication from the BIOS program.

24. (Previously Presented) The method of claim 22, wherein receiving the indication from the BIOS program comprises resetting the configurable time in the watchdog timer.

25. (Previously Presented) The method of claim 20, wherein the event data is stored in a memory of the computer system by a controller device included in the computer system.

26. (Previously Presented) The method of claim 25, wherein the controller device is a memory controller.

27. (Previously Presented) The method of claim 25, wherein the controller device is an I/O controller.

28. (Previously Presented) The method of claim 20, wherein the system management controller accesses the event data over a system bus of the computer system.

29. (Previously Presented) The method of claim 28, wherein the system bus is a SMBus.

30. (Previously Presented) The method of claim 20 further comprising, the

system management controller writing the event log in response to accessing the event data.

31. (Previously Presented) The method of claim 20, wherein writing the event log occurs over a system bus of the computer system.

32. (Previously Presented) The method of claim 31, wherein the system bus is a SMBus.